

AMENDMENTS TO THE CLAIMS

1-8. (Cancelled)

9. (Currently amended) ~~The network system of claim 5,~~ A network interface that processes data cells transmitted on a network operating in asynchronous transfer mode (ATM), comprising:

a data path unit that inputs said data cells transmitted on said network;

a header processor that inputs a first data cell of said data cells from said data path unit and determines whether or not a full tuple can be created based on said first data cell; and

a pointer memory that contains a pointer and a validity indicator,

wherein said pointer points to another memory location which stores information corresponding to at least a portion of said full tuple and said validity indicator identifies a characteristic of said pointer,

wherein, if said full tuple can be created based on said first data cell, said header processor constructs said full tuple based on said first data cell and outputs said full tuple,

wherein, if said full tuple cannot be created based on said first data cell, said header processor inputs a second data cell of said data cells from said data path unit, constructs said full tuple based on said first data cell and said second data cell, and outputs said full tuple,

wherein said data path unit creates a data packet corresponding to said full tuple,

wherein, when said validity indicator corresponds to valid pointer data, said pointer points to a first memory location when said valid pointer data has a first value and points to a second memory location when said valid pointer data has a second value, and

wherein said second memory location contains process flow information corresponding to a process flow of said data packet.

10. (Currently amended) The network system interface of claim 9, wherein said process flow information is stored in said second memory location and said valid pointer data has said second value after said full tuple has been created.

11-16. (Canceled)

17. (Original) A method for classifying Internet protocol (IP) packets transferred in data cells over an asynchronous transfer mode (ATM) network, comprising:

(a) determining if a first data cell contains a full IP tuple;

(b) creating said full IP tuple from said first data cell if said first data cell contains said full IP tuple;

(c) classifying said full IP tuple in accordance with process flow information after said full IP tuple is created;

(d) determining a first pointer based on a virtual channel identifier/virtual path identifier (VCI/VPI) contained in said first data cell;

(e) storing a second pointer and validity data in a first memory location, wherein said first pointer points to said first memory location;

(f) storing said process flow information in a second memory location, wherein said second pointer points to said second memory location after said full IP tuple is created;

(g) obtaining an IP packet corresponding to said full IP tuple; and

(h) determining operations to be performed on said IP packet based on said process flow information.

18. (Original) The method of claim 17, further comprising:

(i) storing a payload of said first data cell in a third memory location if said first data cell does not contain said full IP tuple.

19. (Original) The method of claim 18, further comprising:

(j) when said first data cell does not contain said full IP tuple, pointing said second pointer to said third memory location after receiving said first data cell and before said full IP tuple is created; and

(k) when said first data cell does not contain said full IP tuple, setting said validity data to indicate that said second pointer points to said third memory location after receiving said first data cell and before said full IP tuple is created.

20. (Original) The method of claim 19, further comprising:

(l) reading a second data cell if said first data cell does not contain said full IP tuple, wherein, said operation (l) is performed after said operation (j).

21. (Original) The method of claim 20, further comprising:

(m) creating said full IP tuple from at least said first data cell and said second data cell when said first data cell does not contain said full IP tuple.

22. (Original) The method of claim 17, further comprising:

(i) setting said validity indicator to indicate invalidity of said second pointer when a last data cell corresponding to said IP packet is received.

23. (Original) The method of claim 21, further comprising:

(n) setting said validity indicator to indicate invalidity of said second pointer when a last data cell corresponding to said IP packet is received.

24. (Original) The method of claim 17, further comprising:

(i) setting said validity indicator to indicate that said second pointer points to said second memory location after said first tuple is created.

25. (Original) A method for classifying Internet protocol (IP) packets transferred in data cells over an asynchronous transfer mode (ATM) network, comprising:

- (a) receiving a first ATM cell;
- (b) determining if said first ATM cell comprises a full IP tuple;
- (c) if said first ATM cell comprises said full IP tuple, creating said full IP tuple from said first ATM cell;

(d) determining if said full tuple corresponds to an existing process flow;

(e) if said full tuple does not correspond to an existing process flow, generating new process flow information as particular process flow information; and

(f) if said full tuple corresponds to an existing process flow, determining that existing process flow information corresponds to said particular process flow information.

26. (Original) The method as claimed in claim 25, wherein said operation (c) comprises:

(c1) if said first ATM cell comprises said full IP tuple, creating said full IP tuple from said first ATM cell;

(c2) if said first ATM cell does not comprise said full IP tuple, receiving a second ATM cell; and

(c3) after receiving said second ATM cell, if said first ATM cell and said second ATM cell comprise said full IP tuple, creating said full IP tuple from said first ATM cell and said second ATM cell.

27. (Original) The method as claimed in claim 25, further comprising:

(g) storing said particular process flow information in a first memory location,

(h) creating a first pointer from a virtual channel identifier/virtual path identifier (VCI/VPI) contained in said first ATM cell, wherein said first pointer at least indirectly points to said first memory location.

28. (Original) The method as claimed in claim 27, further comprising:

(i) storing a second pointer in a second memory location,

wherein said second pointer comprises a validity field and points to said first memory location after said full IP tuple is created, and

wherein said first pointer points to said second memory location.

29. (Original) The method as claimed in claim 28, wherein said validity field indicates that said second pointer points to said particular process flow information after said full IP tuple is created, and

wherein said validity field indicates that said second pointer is invalid after a last data packet corresponding to said full IP tuple has been received.

30. (Original) The method as claimed in claim 26, further comprising:

(g) when said first ATM cell does not comprise said full IP tuple, storing information corresponding to said first ATM cell in a first memory location after said first ATM cell is received and before said full IP tuple is created,

(h) creating a first pointer from a virtual channel identifier/virtual path identifier (VCI/VPI) contained in said first ATM cell corresponding to said full IP tuple, wherein said first pointer at least indirectly points to said first memory location after said first ATM cell is received and before said full IP tuple is created.

31. (Original) The method as claimed in claim 30, further comprising:

(i) storing a second pointer in a second memory location,

wherein said second pointer comprises a validity field and points to said first memory location after said first ATM cell is received and before said full IP tuple is created,

wherein said first pointer points to said second memory location, and

wherein, after said first ATM cell is received and before said full IP tuple is created, said validity field indicates that said first memory location comprises said information corresponding to said first ATM cell.

32. (Original) The method as claimed in claim 31, further comprising:

(j) storing said particular process flow information in a third memory location,

wherein said validity field indicates that said second pointer points to said third memory location after said full IP tuple is created.

33. (Original) The method as claimed in claim 32, wherein said validity field indicates that said second pointer is invalid after a last data packet corresponding to said full IP tuple has been received.

34. (Original) Software contained in a computer readable medium, wherein said software comprises instructions to instruct a processor for performing operations, comprising:

(a) determining if a first data cell in an asynchronous transfer mode (ATM) network contains a full Internet protocol (IP) tuple;

(b) creating said full IP tuple from said first data cell if said first data cell contains said full IP tuple;

- (c) classifying said full IP tuple in accordance with process flow information after said full IP tuple is created;
- (d) determining a first pointer based on a virtual channel identifier/virtual path identifier (VCI/VPI) contained in said first data cell;
- (e) storing a second pointer and validity data in a first memory location, wherein said first pointer points to said first memory location;
- (f) storing said process flow information in a second memory location, wherein said second pointer points to said second memory location after said full IP tuple is created;
- (g) obtaining an IP packet corresponding to said full IP tuple; and
- (h) determining operations to be performed on said IP packet based on said process flow information.

35. (Original) The software of claim 34, wherein said operations further comprise:

- (i) storing a payload of said first data cell in a third memory location if said first data cell does not contain said full IP tuple.

36. (Original) The software of claim 35, wherein said operations further comprise:

- (j) when said first data cell does not contain said full IP tuple, pointing said second pointer to said third memory location after receiving said first data cell and before said full IP tuple is created; and
- (k) when said first data cell does not contain said full IP tuple, setting said validity data to indicate that said second pointer points to said third memory

location after receiving said first data cell and before said full IP tuple is created.

37. (Original) The software of claim 36, wherein said operations further comprise:

(l) reading a second data cell if said first data cell does not contain said full IP tuple, wherein, said operation (l) is performed after said operation (j).

38. (Original) The software of claim 37, wherein said operations further comprise:

(m) creating said full IP tuple from at least said first data cell and said second data cell when said first data cell does not contain said full IP tuple.

39. (Original) The software of claim 34, wherein said operations further comprise:

(i) setting said validity indicator to indicate invalidity of said second pointer when a last data cell corresponding to said IP packet is received.

40. (Original) The software of claim 38, wherein said operations further comprise:

(n) setting said validity indicator to indicate invalidity of said second pointer when a last data cell corresponding to said IP packet is received.

41. (Original) The software of claim 34, wherein said operations further comprise:

(i) setting said validity indicator to indicate that said second pointer points to said second memory location after said first tuple is created.

42. (Original) Software contained in a computer readable medium, wherein said software comprises instructions to instruct a processor for performing operations, comprising:

- (a) receiving a first ATM cell;
- (b) determining if said first ATM cell comprises a full IP tuple;
- (c) if said first ATM cell comprises said full IP tuple, creating said full IP tuple from said first ATM cell;
- (d) determining if said full tuple corresponds to an existing process flow;
- (e) if said full tuple does not correspond to an existing process flow, generating new process flow information as particular process flow information; and
- (f) if said full tuple corresponds to an existing process flow, determining that existing process flow information corresponds to said particular process flow information.

43. (Original) The software as claimed in claim 42, wherein said operation (c) comprises:

- (c1) if said first ATM cell comprises said full IP tuple, creating said full IP tuple from said first ATM cell;
- (c2) if said first ATM cell does not comprise said full IP tuple, receiving a second ATM cell; and
- (c3) after receiving said second ATM cell, if said first ATM cell and said second ATM cell comprise said full IP tuple, creating said full IP tuple from said first ATM cell and said second ATM cell.

44. (Original) The software as claimed in claim 42, wherein said operations further comprise:

(g) storing said particular process flow information in a first memory location,

(h) creating a first pointer from a virtual channel identifier/virtual path identifier (VCI/VPI) contained in said first ATM cell, wherein said first pointer at least indirectly points to said first memory location.

45. (Original) The software as claimed in claim 44, wherein said operations further comprise:

(i) storing a second pointer in a second memory location,

wherein said second pointer comprises a validity field and points to said first memory location after said full IP tuple is created, and

wherein said first pointer points to said second memory location.

46. (Original) The software as claimed in claim 45, wherein said validity field indicates that said second pointer points to said particular process flow information after said full IP tuple is created, and

wherein said validity field indicates that said second pointer is invalid after a last data packet corresponding to said full IP tuple has been received.

47. (Original) The software as claimed in claim 43, wherein said operations further comprise:

(g) when said first ATM cell does not comprise said full IP tuple, storing information corresponding to said first ATM cell in a first memory location after said first ATM cell is received and before said full IP tuple is created,

(h) creating a first pointer from a virtual channel identifier/virtual path identifier (VCI/VPI) contained in

said first ATM cell corresponding to said full IP tuple, wherein said first pointer at least indirectly points to said first memory location after said first ATM cell is received and before said full IP tuple is created.

48. (Original) The software as claimed in claim 47, wherein said operations further comprise:

(i) storing a second pointer in a second memory location,

wherein said second pointer comprises a validity field and points to said first memory location after said first ATM cell is received and before said full IP tuple is created,

wherein said first pointer points to said second memory location, and

wherein, after said first ATM cell is received and before said full IP tuple is created, said validity field indicates that said first memory location comprises said information corresponding to said first ATM cell.

49. (Original) The software as claimed in claim 48, wherein said operations further comprise:

(j) storing said particular process flow information in a third memory location,

wherein said validity field indicates that said second pointer points to said third memory location after said full IP tuple is created.

50. (Original) The software as claimed in claim 49, wherein said validity field indicates that said second pointer is invalid after a last data packet corresponding to said full IP tuple has been received.